

Oracle Database In-Memory

Enabling Real-Time Analytics – The Details

Andy Rivenes

Database In-Memory Product Management
Oracle Corporation

Email: andy.rivenes@oracle.com

Twitter: @TheInMemoryGuy

Blog: blogs.oracle.com/in-memory



ORACLE

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Database Platform of the **Future** – *Complete and Integrated*

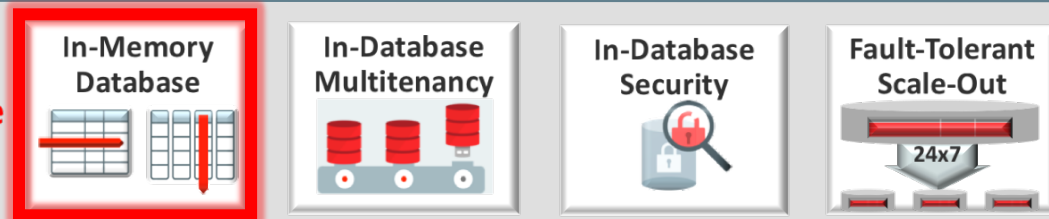
Services

Any Application
Any Data



Architecture

Fast, Low Cost, Secure
Scalable, Available



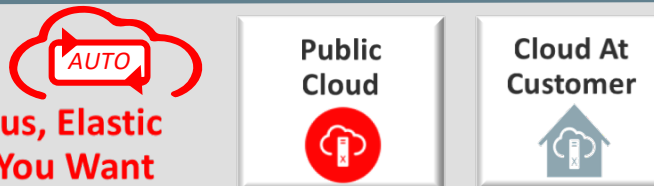
Systems

Engineered
for Databases



Cloud

Autonomous, Elastic
Wherever You Want



ORACLE®

Copyright © 2018, Oracle and/or its affiliates. All rights reserved. |

3

What is Database In-Memory



Oracle Database In-Memory

Real-Time Analytics



Enable Real-Time
Business Decisions

Accelerate Mixed Workload



Run analytics on
Operational
Systems

Risk-Free



Proven Scale-Out,
Availability, Security

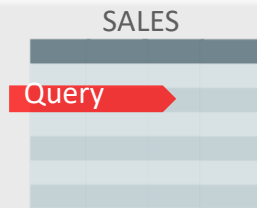
Trivial to Implement



No Application Changes
Not Limited by Memory

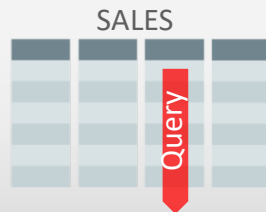
Row Format Databases vs. Column Format Databases

Rows Stored
Contiguously



- **Transactions** run faster on row format
 - Example: Query or Insert a sales order
 - Fast processing few rows, many columns

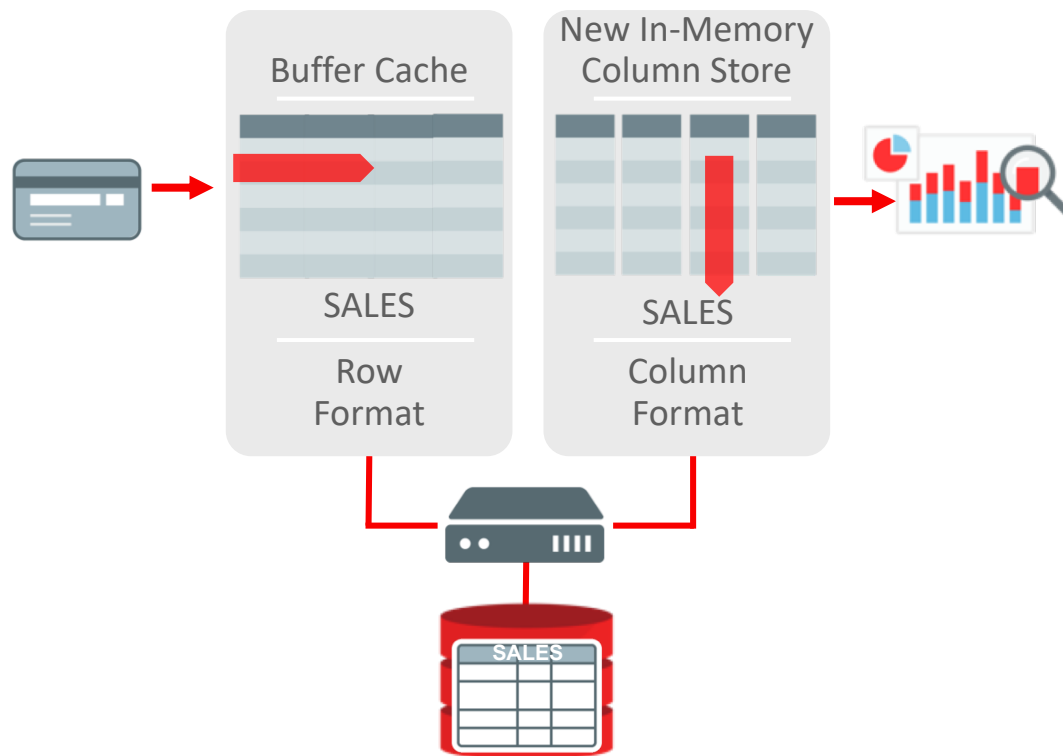
Columns Stored
Contiguously



- **Analytics** run faster on column format
 - Example : Report on sales totals by region
 - Fast accessing few columns, many rows

Until Now Must Choose One Format and Suffer Tradeoffs

Breakthrough: Dual Format Database



- **BOTH** row and column formats for same table
- Simultaneously active and transactionally consistent
- Analytics & reporting use new in-memory Column format
- OLTP uses proven row format

The Forrester Wave™: In-Memory Databases, Q1 2017

- Oracle In-Memory Scored Highest of All Databases for Current Offering and Strategy
- For the full report see:
<https://reprints.forrester.com/#/assets/2/132/%27RES132143%27/reports>

Where Is It Available



Oracle Database

- Database In-Memory is an option for Oracle Database 12c Enterprise Edition
- Database In-Memory is included in the first patchset (12.1.0.2) for 12.1
- Oracle Database 12c Release 2 (12.2) is available in the cloud and on-premises
- Will also be available in Oracle Database 18c as it is released in the Cloud, Exadata and on-premises
- No additional installation steps required



Note: Database In-Memory is **not** enabled by default

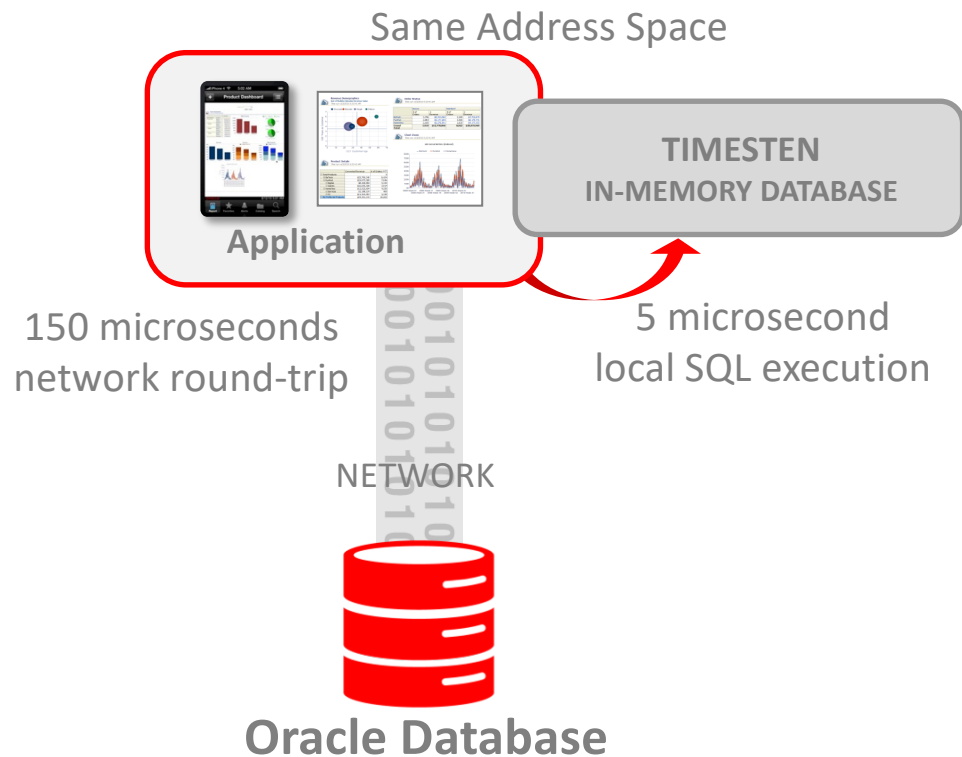
Where Database In-Memory Is Available

- Exadata Express Cloud Service – X1000IM – up to 10GB Column Store
- Database Cloud Service: **Extreme Performance**
- Exadata Cloud Service
- Exadata Cloud at Customer
- On-premises

Isn't it just TimesTen



TimesTen for Latency-Critical OLTP Complementary In-Memory Technology

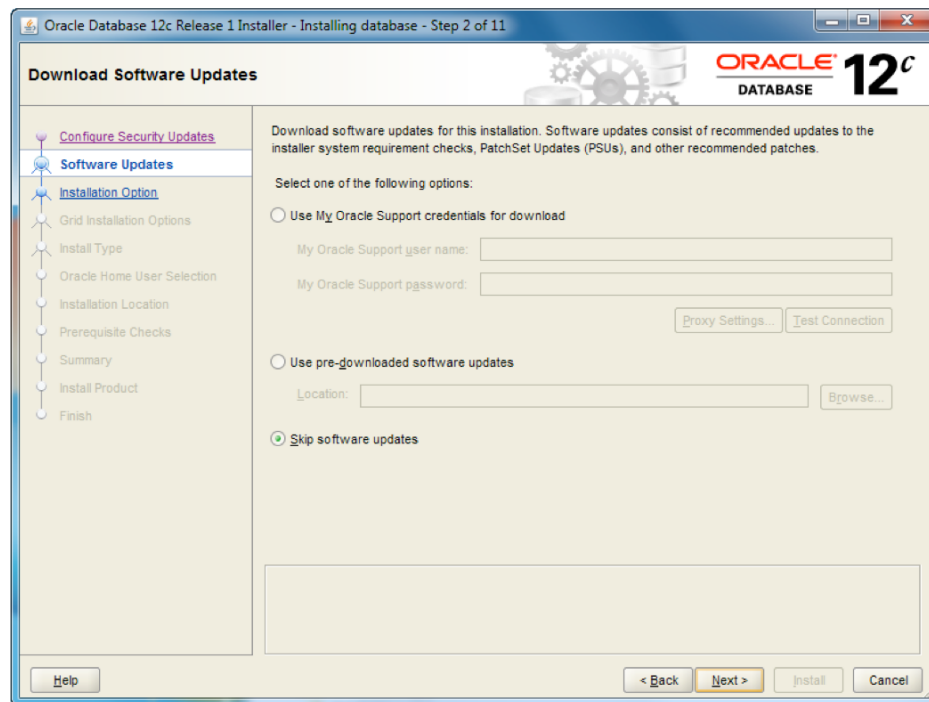


- **Latency-Critical OLTP limited by network between application and database**
 - Phone call routing, stock trading
- TimesTen In-Memory Database is light-weight and ultra-fast
 - Runs in application address space:
No Network
 - **30x** faster latency-critical OLTP

How easy is it to get started



Installing : Oracle Database In-Memory



Automatically installed as
part of Database 12c
(12.1.0.2 & 12.2)

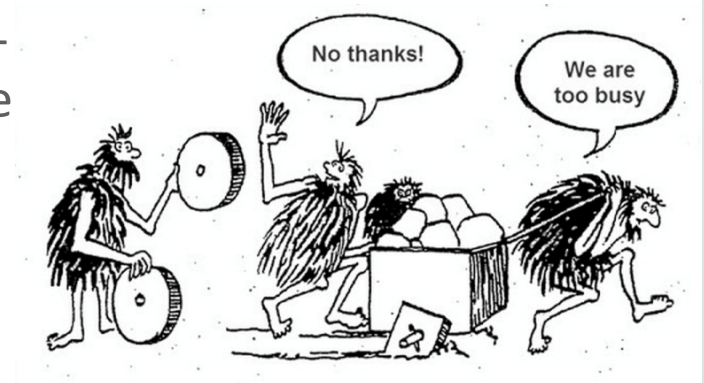
No additional steps
required



Note: Database In-Memory is **not** enabled by default

Installing: Apply the Latest Database Proactive Bundle Patch

- Database In-Memory fixes and enhancements are **only** distributed through Database Proactive Bundle Patches
- See MOS Notes:
 - 2285557.1 – Database 12.2.0.1 Proactive Patch Information
 - 2285558.1 – Database 12.1.0.2 Proactive Patch Information
- Starting with the latest bundle patch avoids re-inventing the wheel discovering bugs that have already been fixed!



Oracle In-Memory: Simple to Implement

1. Configure Memory Capacity

- `inmemory_size = XXX GB`

2. Configure tables or partitions to be in memory

- `alter table | partition ... inmemory;`

3. Later drop analytic indexes to speed up OLTP

Oracle In-Memory Advisor

Object Type	Object	Estimated In-Memory Size	Analytics Processing Seconds	Estimated Reduced Analytics Processing Seconds	Estimated Analytics Processing Performance Improvement Factor	Benefit / Cost Ratio (Improvement Factor / In-Memory Size)
Table	SOE.LOGON	451.76MB	2114	1,887	9.3X	20.586
Table	SOE.CARD_DETAILS	607.32MB	8346	7,248	7.6X	12.514
Table	SOE.ADDRESSES	1.09GB	5237	4,621	8.5X	7.798
Partition	SOE.PRODUCT MOCKUP.Y2014Q1	812.6MB	2003	1,489	3.9X	4.799
Table	SOE.CUSTOMERS	1.10GB	108	95	8.2X	7.455
Table	SOE.ORDER_ITEMS	2.19GB	7128	6,393	9.7X	4.429
Table	SOE.ORDERS	1.34GB	3512	2,917	5.9X	4.403
Table	SOE.PRODUCT_INFORMATION	1.78MB	2873	2,205	4.3X	2.416
Partition	SOE.PRODUCT MOCKUP.Y2013Q4	1.62GB	97	1,489	3.7X	2.284
Partition	SOE.PRODUCT MOCKUP.Y2014Q2	3.37GB	642	493	4.3X	1.276

- New In-Memory Advisor
- Analyzes existing DB workload via AWR & ASH repositories
- Provides list of objects that would benefit most from being populated into IM column store



Note: Database Tuning Pack license required

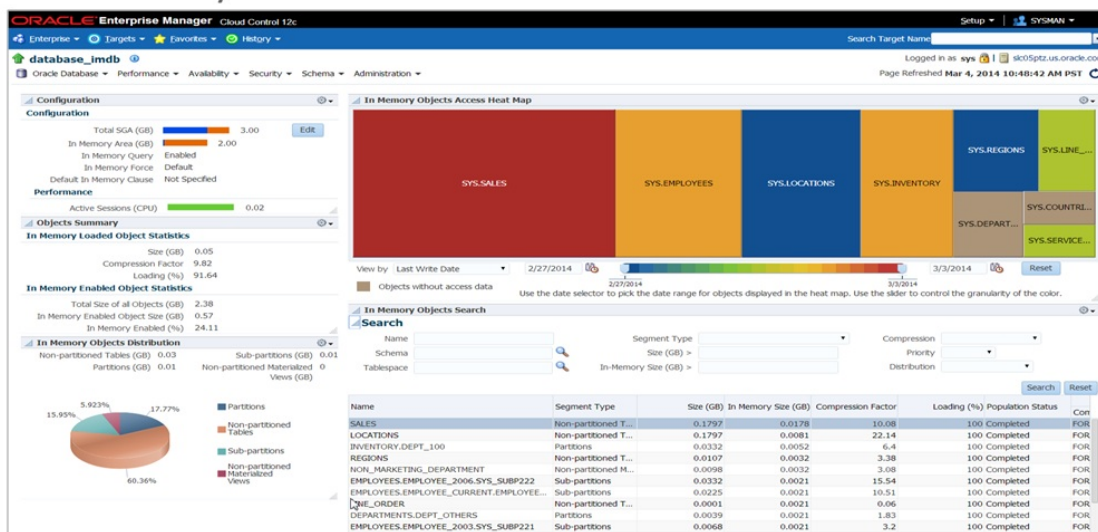
Oracle Compression Advisor And In-Memory

```
DECLARE
  l_blkcnt cmp          PLS_INTEGER;
  l_blkcnt uncmp        PLS_INTEGER;
  l_row cmp             PLS_INTEGER;
  l_row uncmp           PLS_INTEGER;
  cmp_ratio             PLS_INTEGER;
  l_comptype str        VARCHAR2(100);
  comp_ratio_allrows NUMBER := -1;
BEGIN
  dbms_compression.Get_compression_ratio (
    scratchtbsname => 'TS_DATA',
    ownname        => 'SSB',
    objname        => 'LINEORDER',
    subobjname     => NULL,
    comptype       => dbms_compression.comp_inmemory_query_low,
    blkcnt cmp     => l_blkcnt cmp,
    blkcnt uncmp   => l_blkcnt uncmp,
    row cmp        => l_row cmp,
    row uncmp      => l_row uncmp,
    cmp_ratio      => cmp_ratio,
    comptype str   => l_comptype str,
    subset numRows => dbms_compression.comp_ratio_allrows);
  dbms_output.Put_line('The IM compression ratio is ' || cmp_ratio);
END;
```

- Easy way to determine memory requirements
- Use DBMS_COMPRESSION
- Applies MEMCOMPRESS to sample set of data from a table
- Returns estimated compression ratio

Oracle Enterprise Manager: In-Memory Central

In-Memory Central



- OEM supports Database In-Memory
- In-Memory Central page gives a dashboard look to the IM column store
- Provides list of objects populated in the IM column store

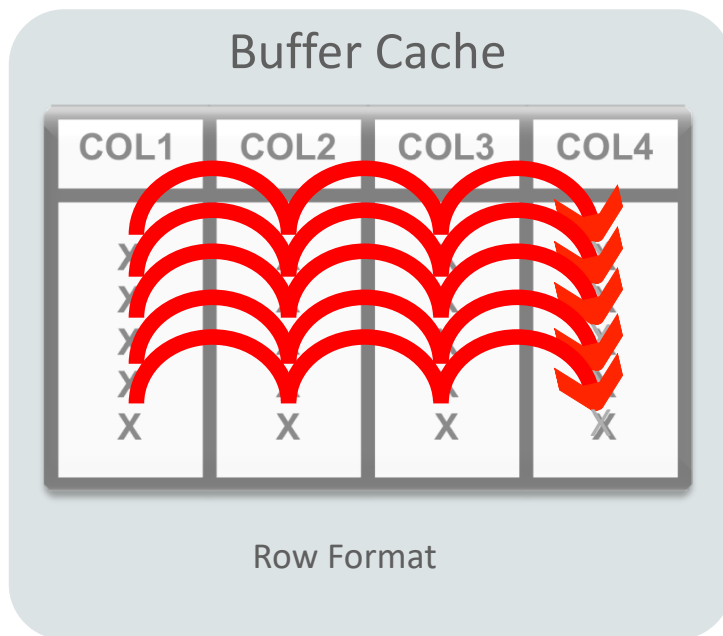
That's all you need to know, but wait ...



How does it work



Why is an In-Memory scan faster than the buffer cache?



SELECT **COL4** FROM MYTABLE;



RESULT

Why is an In-Memory scan faster than the buffer cache?

IM Column Store

COL1	COL2	COL3	COL4
X	X	X	X
X	X	X	X
X	X	X	X
X	X	X	X
X	X	X	X

Column Format

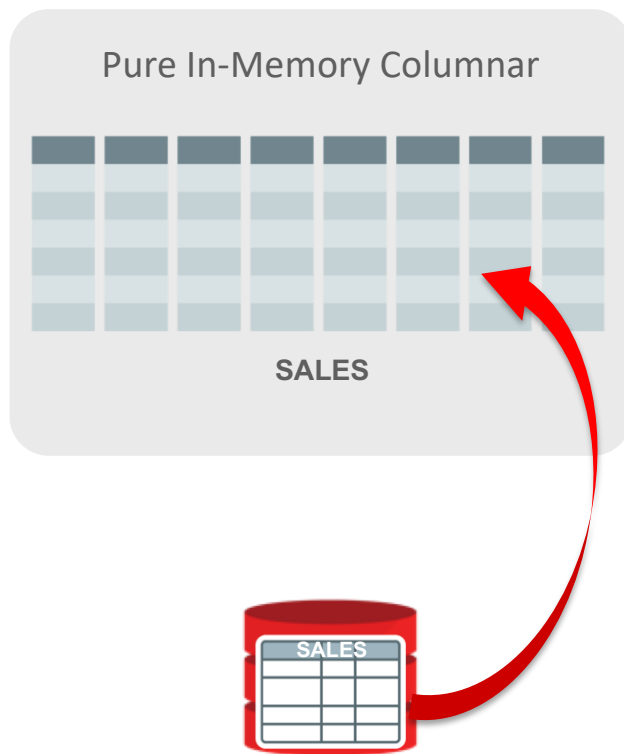
X
X
X
X
X

SELECT **COL4** FROM MYTABLE;



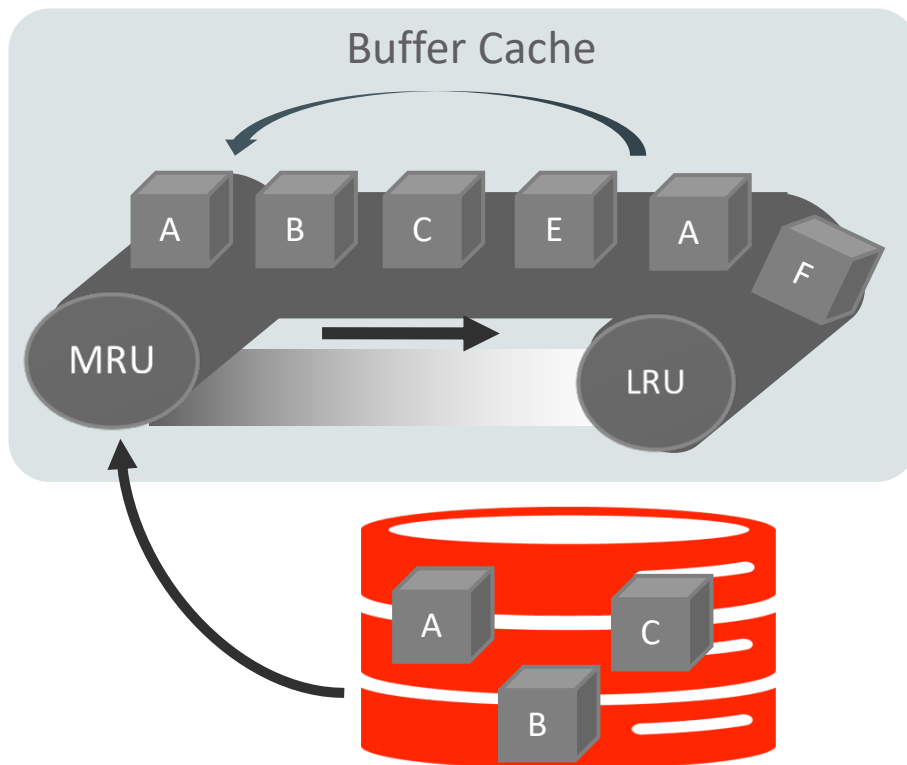
RESULT

Oracle In-Memory Columnar Technology



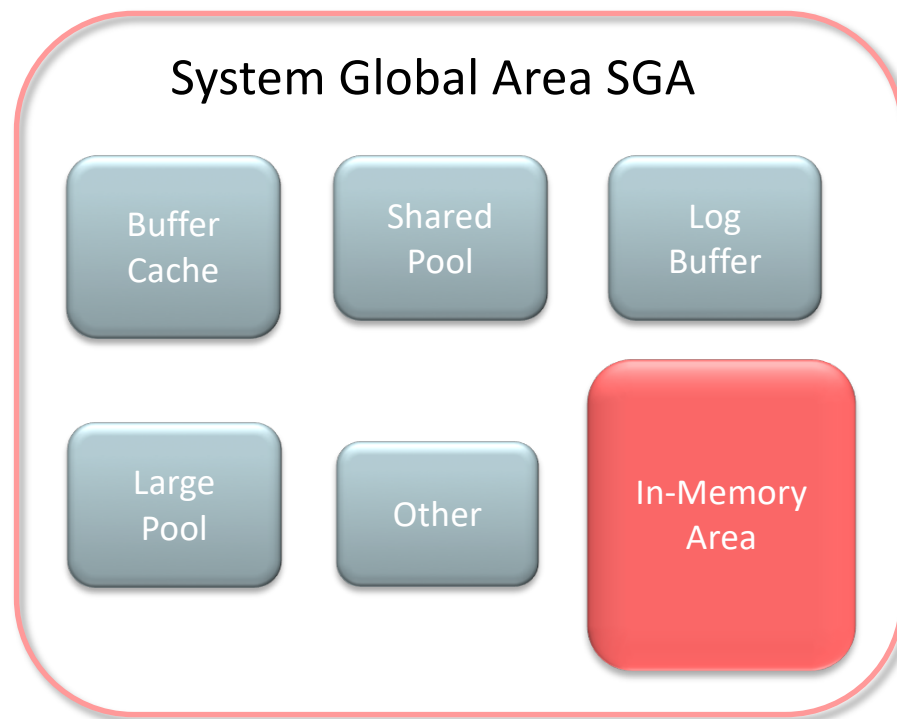
- Pure in-memory column format
 - Enable for subset of database
 - Cheap to maintain – no logging or IO
 - Allows efficient OLTP
 - No change to disk format
- Built **seamlessly** into Oracle Database
 - Appears as a new storage type
 - **Transparent** to Applications
 - All Enterprise Features work ..
 - Availability – RAC, Flashback, DataGuard, etc.
 - Security – Encryption, Auditing, etc.

In-Memory A Store – **Not A Cache**



- What is a cache?
- A pool of memory
- Data automatically brought into memory based on access
- Data automatically aged out
- Good example:
Oracle Database Buffer Cache

In-Memory Area: **New Static Area within SGA**



- Contains data in the new In-Memory Column Format
- Controlled by INMEMORY_SIZE parameter
 - Minimum size of 100MB
- SGA_TARGET must be large enough to accommodate this area

Configuring : Dynamic In-Memory Column Store

```
ALTER SYSTEM SET  
inmemory_size = 300m scope=both;
```

```
SQL> SELECT *  
2 FROM v$inmemory_area;
```

POOL	ALLOC_BYTES	USED_BYTES	POPULATE_STATUS	CON_ID
1MB POOL	124780544	102760448	DONE	0
64KB POOL	16777216	851968	DONE	0

```
SQL>  
SQL> ALTER SYSTEM SET inmemory_size = 300M;
```

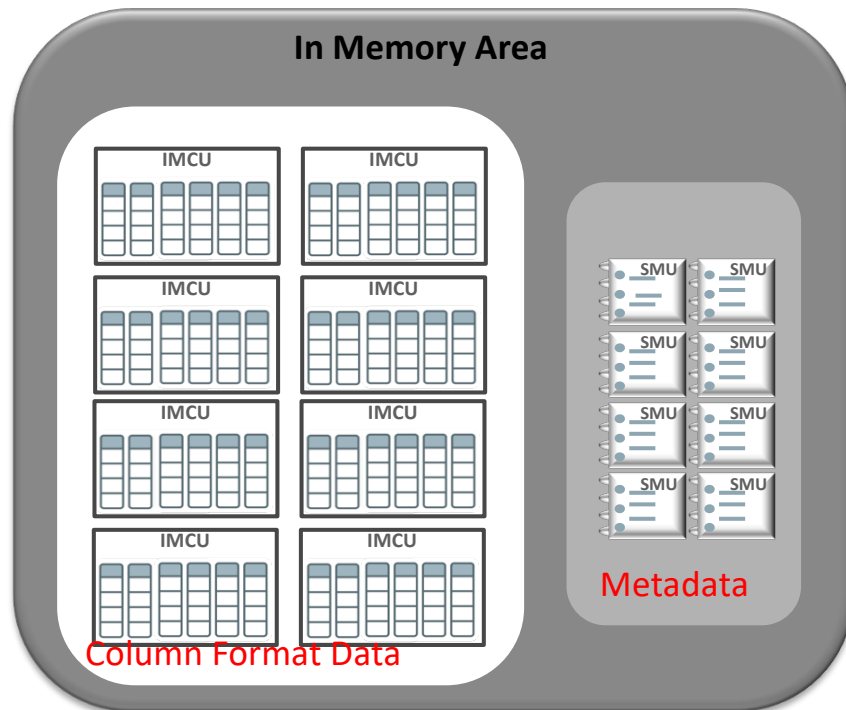
System altered.

```
SQL>  
SQL> SELECT *  
2 FROM v$inmemory_area;
```

POOL	ALLOC_BYTES	USED_BYTES	POPULATE_STATUS	CON_ID
1MB POOL	216006656	102760448	DONE	0
64KB POOL	32374784	851968	DONE	0

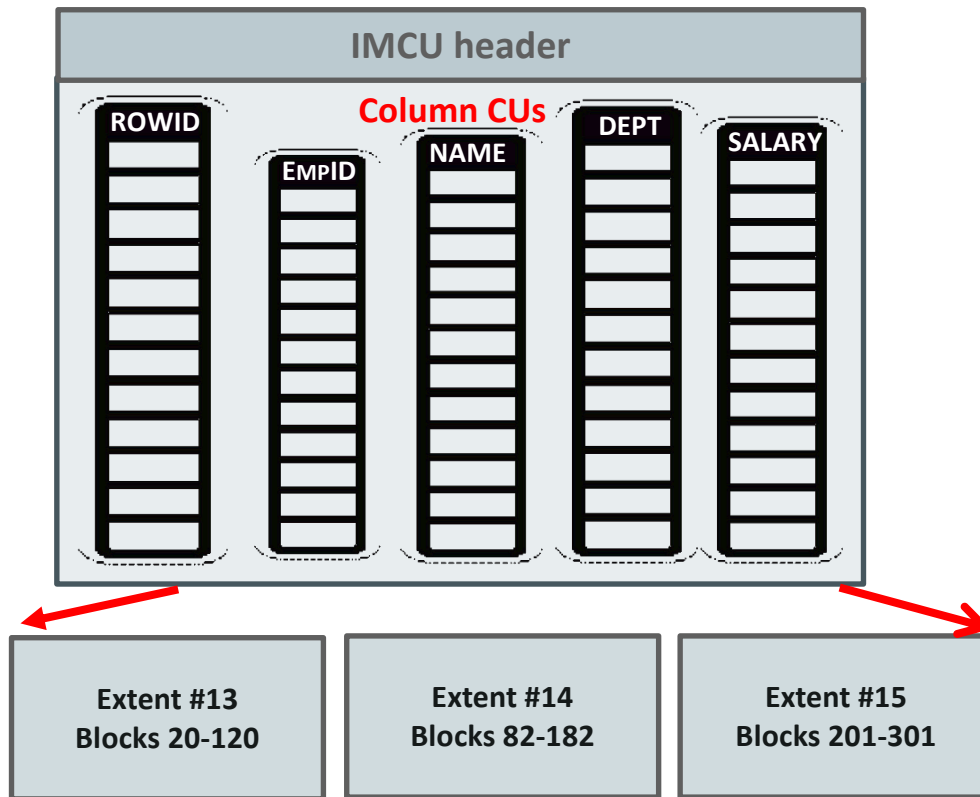
- IM column store is now dynamic
 - Possible to increase the size of IM column store without database restart
 - The IM column store **can not** shrink
 - Only possible if free memory available within the SGA
 - Only possible if new size is **128MB** greater than current INMEMORY_SIZE
 - Change reflected immediately in V\$INMEMORY_AREA

Composition of In-Memory Area



- Contains two subpools:
 - IMCU pool: Stores In Memory Compression Units (IMCUs)
 - SMU pool: Stores Snapshot Metadata Units (SMUs)
- IMCUs contain column formatted data
- SMUs contain metadata and transactional information

Composition of In-Memory Compression Unit (IMCU)



- Unit of column store allocation
 - Columnar representation of a large number of rows from an object
 - Rows from one or more table extents
- Actual size depends on size of rows, compression factor, etc.
- Each column stored as a separate contiguous **Column Compression Unit** (column CU)
 - Rowids also stored as a Column CU

Composition of In-Memory Area

- V\$INMEMORY_AREA:
Current sizes of pools and pool statuses
- V\$IM_HEADER: List of IMCUs currently in the inmemory column store

```
SQL> SELECT * from V$INMEMORY_AREA;
```

POOL	ALLOC_BYTES	USED_BYTES	POPULATE_STATUS
1MB POOL	1307574272	135266304	DONE
64KB POOL	318767104	524288	DONE

```
SQL> SELECT OBJD, TSN, ALLOCATED_LEN, NUM_ROWS, NUM_COLS  
FROM V$IM_HEADER;
```

OBJD	TSN	ALLOCATED_LEN	NUM_ROWS	NUM_COLS
92918	5	19922944	532246	60
92918	5	19922944	532480	60
92918	5	19922944	522496	60
92918	5	11534336	532480	60
92918	5	19922944	530176	60
92918	5	19922944	532480	60
92918	5	15728640	419562	60



How Do I Get Data In And Out Of The In-Memory Column Store?

Populating : Enable Objects for In-Memory

```
ALTER TABLE sales INMEMORY;
```

```
ALTER TABLE sales NO INMEMORY;
```

```
CREATE TABLE customers .....  
PARTITION BY LIST  
  (PARTITION p1 ..... INMEMORY,  
   (PARTITION p2 ..... NO INMEMORY) ;
```

- New INMEMORY ATTRIBUTE
- Eligible segment types are
 - Tables
 - Partitions
 - Subpartitions
 - Materialized views
- Following types not eligible
 - IOTs
 - Hash clusters
 - Out of line LOBs

Pure OLTP
Features

Populating : Columns Can Be Excluded

```
ALTER TABLE sales INMEMORY  
NO INMEMORY (delivery_note);
```

- You don't have to populate all columns
- It is possible to populate only certain columns
- Two phase approach
 1. INMEMORY attribute on Table automatically inherited by columns
 2. Need to remove attribute from the columns you don't want populated

Populating : Compression

```
ALTER MATERIALIZED VIEW mv1  
INMEMORY  
MEMCOMPRESS FOR QUERY LOW;
```

```
CREATE TABLE trades  
  (Name varchar(20) ,  
   Desc varchar(200))  
INMEMORY  
MEMCOMPRESS FOR DML(desc) ;
```

- Objects compressed during population
- New compression techniques
 - Focused on scan performance
- 2x to 20x compression typical
- Multiple levels of compression
- Possible to use a different level for different partitions in a table

Populating : Compression

```
CREATE TABLE ORDERS .....  
PARTITION BY RANGE .....  
  (PARTITION p1 .....  
    INMEMORY MEMCOMPRESS FOR CAPACITY HIGH  
  PARTITION p2 .....  
    INMEMORY MEMCOMPRESS FOR QUERY LOW,  
  PARTITION p3 .....  
    INMEMORY MEMCOMPRESS FOR DML,  
  :  
  PARTITION p200 .....  
    NO INMEMORY  
  ) ;
```

- Different levels
 - FOR DML
Use on tables or partitions with very active DML activity
 - FOR QUERY LOW / HIGH
Default mode for most tables
 - FOR CAPACITY LOW / HIGH
For less frequently accessed segments
- Easy to switch levels as part of ILM strategy

Populating : Priority

- Order in which objects are populated controlled by PRIORITY subclause:

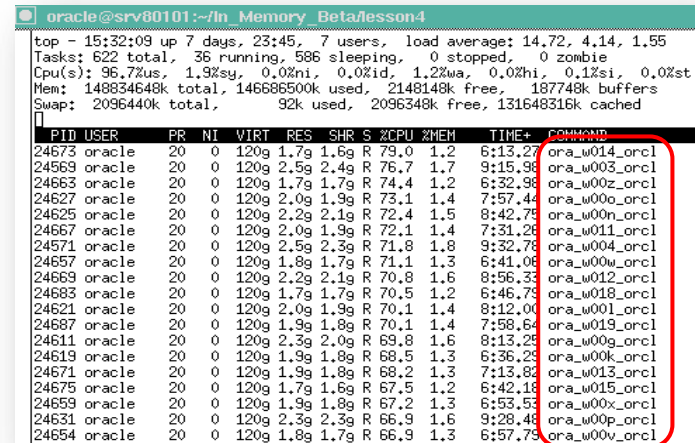
```
ALTER TABLE sales  
INMEMORY PRIORITY HIGH;
```

- Levels:
 - CRITICAL > HIGH > MEDIUM > LOW
 - Controls order (not speed) of populate
- Default PRIORITY is NONE
 - Populate only on first access

- Population completed by new set of background processes

ora_w001_orcl

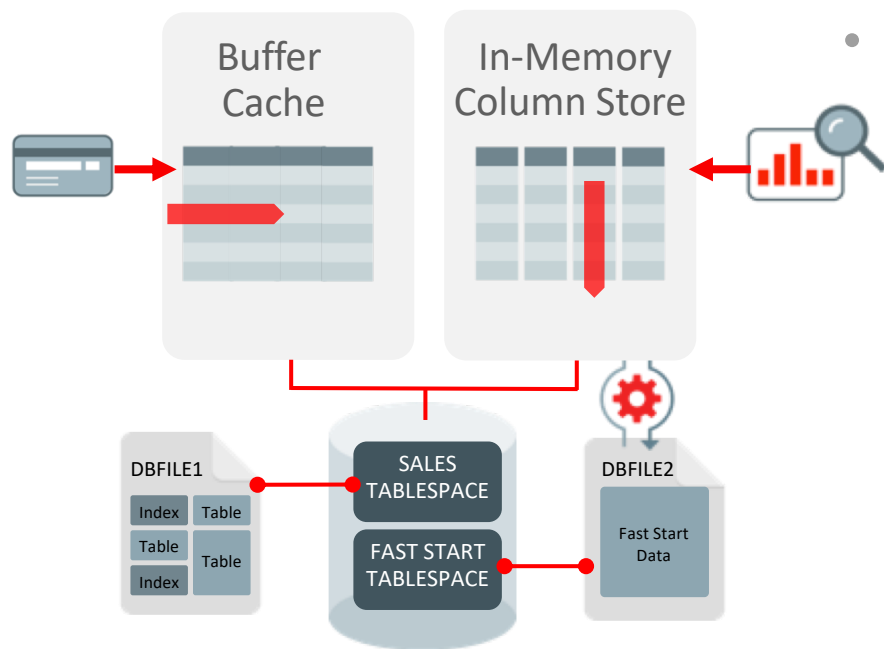
- Number of processes controlled by parameter:
INMEMORY_MAX_POPULATE_SERVERS



```
oracle@srv80101:~/ln_Memory_Beta/lesson4  
top - 15:32:09 up 7 days, 23:45, 7 users, load average: 14.72, 4.14, 1.55  
Tasks: 622 total, 36 running, 586 sleeping, 0 stopped, 0 zombie  
Cpu(s): 96.7%us, 1.9%sy, 0.0%ni, 0.0%id, 1.2%wa, 0.0%hi, 0.1%si, 0.0%st  
Mem: 148834648k total, 146686500k used, 2148148k free, 187748k buffers  
Swap: 2096440k total, 92k used, 2096348k free, 131648316k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
24673	oracle	20	0	120g	1.7g	1.6g	R	79.0	1.2	6:13.27	ora_w014_orcl
24569	oracle	20	0	120g	2.5g	2.4g	R	76.7	1.7	9:15.98	ora_w003_orcl
24663	oracle	20	0	120g	1.7g	1.7g	R	74.4	1.2	6:32.98	ora_w00z_orcl
24627	oracle	20	0	120g	2.0g	1.9g	R	73.1	1.4	7:57.44	ora_w00o_orcl
24625	oracle	20	0	120g	2.2g	2.1g	R	72.4	1.5	8:42.75	ora_w00n_orcl
24667	oracle	20	0	120g	2.0g	1.9g	R	72.1	1.4	7:31.26	ora_w011_orcl
24571	oracle	20	0	120g	2.5g	2.3g	R	71.8	1.8	9:32.78	ora_w004_orcl
24657	oracle	20	0	120g	1.8g	1.7g	R	71.1	1.3	6:41.06	ora_w00w_orcl
24669	oracle	20	0	120g	2.2g	2.1g	R	70.8	1.6	8:56.33	ora_w012_orcl
24683	oracle	20	0	120g	1.7g	1.7g	R	70.5	1.2	6:46.79	ora_w018_orcl
24621	oracle	20	0	120g	2.0g	1.9g	R	70.1	1.4	8:12.00	ora_w001_orcl
24687	oracle	20	0	120g	1.9g	1.8g	R	70.1	1.4	7:58.64	ora_w019_orcl
24611	oracle	20	0	120g	2.3g	2.0g	R	69.8	1.6	8:13.25	ora_w00g_orcl
24619	oracle	20	0	120g	1.9g	1.8g	R	68.5	1.3	6:36.23	ora_w00k_orcl
24671	oracle	20	0	120g	1.9g	1.8g	R	68.2	1.3	7:13.82	ora_w013_orcl
24675	oracle	20	0	120g	1.7g	1.6g	R	67.5	1.2	6:42.18	ora_w015_orcl
24659	oracle	20	0	120g	1.9g	1.8g	R	67.2	1.3	6:53.53	ora_w00x_orcl
24631	oracle	20	0	120g	2.3g	2.3g	R	66.9	1.6	9:28.48	ora_w00p_orcl
24654	oracle	20	0	120g	1.8g	1.7g	R	66.9	1.3	6:57.79	ora_w00v_orcl

Population Performance : In-Memory Fast-Start



- IM column format persisted to storage
- In-Memory column store contents checkpointed to secure file lob on populate
- When DB restarts population is faster as population process reads the column format directly from storage
- Faster restore (**2-5x**) of column store since no need to reformat data

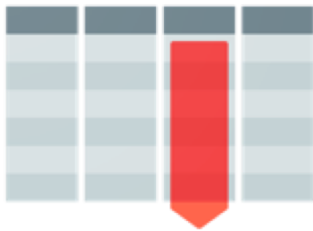


Why Are Analytic Queries Faster In The In-Memory Column Store?

Real-Time Analytics

Scanning and filtering data more efficiently

Columnar Format



Access only the columns you need

Compression



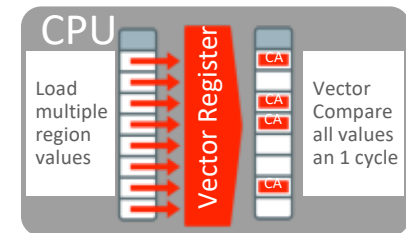
Scan & filter data in compressed format

Storage Indexes



Prune out any unnecessary data from the column

SIMD Vector Processing



Process multiple column values in a single CPU instruction

Real-time Analytics

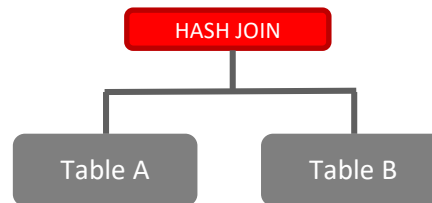
Improves all aspects of analytic queries

Data Scans



- Speed of memory
- Scan and Filter only the needed Columns
- Vector Instructions

Joins



- Convert Star Joins into 10X Faster Column Scans
- Search large table for values that match small table

In-Memory Aggregation



- Create In-Memory Report Outline that is Populated during Fast Scan
- Runs Reports Instantly

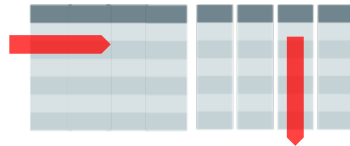
Improvements in 12.2 for Database In-Memory

Real-Time Analytics



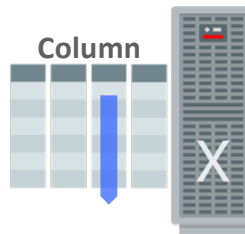
- **2X** Faster Joins
- **5X** Faster Expressions

Mixed Workload



- Active Data Guard Support

Massive Capacity



- In-Memory on Exadata Flash

Multi-model



- Native support for JSON Data type

Automation



- Dynamic Data Movement Between Storage & Memory
- IM FastStart
- IM Column Store Re-sizing

Improvements in 18c for Database In-Memory

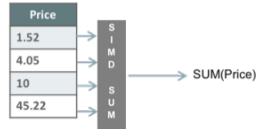
Further Performance Gains



2X Query Performance Gains
Exadata Flash support for non-HCC objects

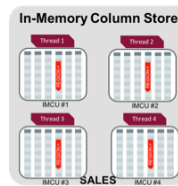
Optimized Arithmetic

FASTER SIMD Vector Processing of In-Memory Numbers



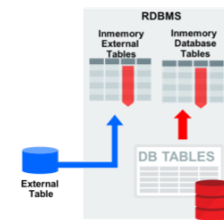
In-Memory Optimized Arithmetic

Dynamic Scans



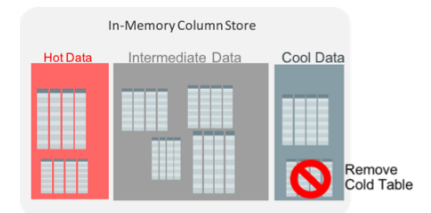
In-Memory Dynamic Scans

External Tables



In-Memory External Tables

Automatic In-Memory



Automatic Data Movement
Between Storage & Memory

Why not just “cache”
the table in the row
store



Compare Column-store to Row-store

```
SQL> -- In-Memory Column Store query
SQL>
SQL> select max(lo_ordtotalprice) most_expensive_order From LINEORDER;
```

MOST_EXPENSIVE_ORDER

57346348

Elapsed: 00:00:00.01

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				5401 (100)	
1	SORT AGGREGATE		1	6		
2	TABLE ACCESS INMEMORY FULL	LINEORDER	59M	343M	5401 (16)	00:00:01

```
SQL> -- Buffer Cache query with the column store disabled via NO_INMEMORY hint
```

```
SQL>
SQL> select /*+ NO_INMEMORY */ max(lo_ordtotalprice) most_expensive_order From LINEORDER;
```

MOST_EXPENSIVE_ORDER

57346348

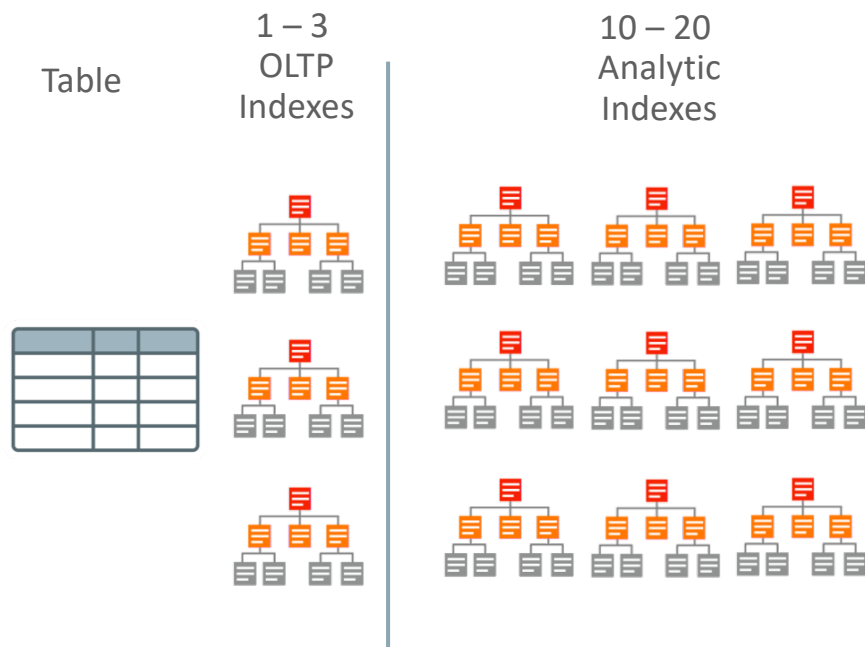
Elapsed: 00:00:08.38

Id	Operation	Name	Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT				123K(100)	
1	SORT AGGREGATE		1	6		
2	TABLE ACCESS FULL	LINEORDER	59M	343M	123K (1)	00:00:05

How does it impact OLTP environments



Complex OLTP is Slowed by Analytic Indexes



- Most Indexes in complex OLTP (e.g. ERP) databases are only used for analytic queries
- Inserting one row into a table requires updating 10-20 analytic indexes: **Slow!**
- Indexes only speed up predictable queries & reports

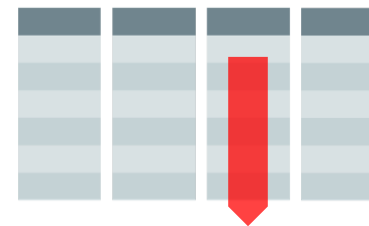
Database In-Memory Accelerates Mixed Workloads

- Complex OLTP is Slowed by Analytic Indexes



- Inserting one row into a table requires updating 10-20 analytic indexes: **Slow!**

- Column Store Replaces Analytic Indexes



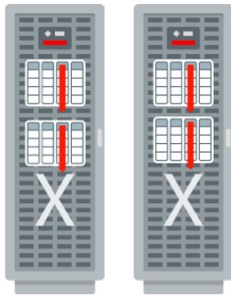
- Fast analytics on any columns
- Column Store not persistent so update cost is much lower

Database In-Memory and other Oracle Database features



Database In-Memory: Scales to Any Size

Scale-Out



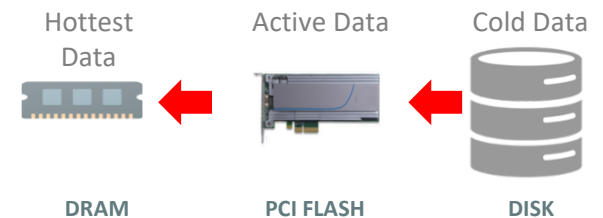
- Scale-Out Across Servers to Grow Memory and CPUs
- In-Memory Queries Parallelized Across Servers

Scale-Up



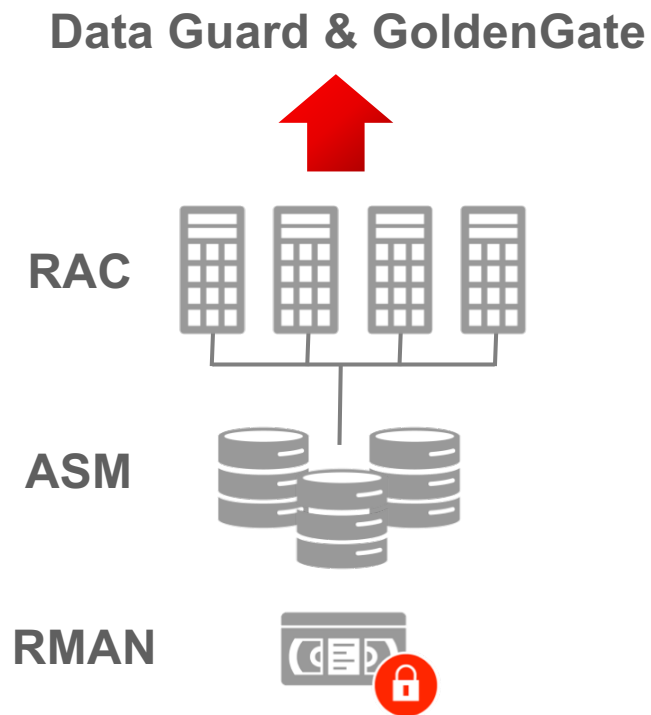
- Scale-Up on large SMPs
- NUMA Optimized

Combine with Flash and Disk



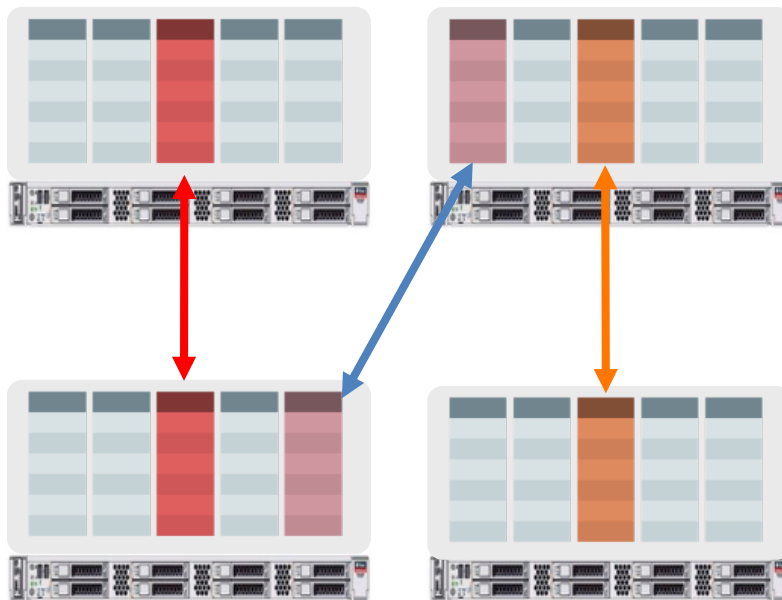
- Easily place data on most cost effective tier
- Simultaneously Achieve:
 - **Speed** of DRAM
 - **I/Os** of Flash
 - **Cost** of Disk

Database In-Memory: **Industrial Strength Availability**



- Pure In-Memory format does not change Oracle's storage format, logging, backup, recovery, etc.
- All Oracle's proven availability technologies work transparently
- **Protection from all failures**
 - Node, site, corruption, human error, etc.

Database In-Memory: Unique Fault Tolerance



Only Available on Engineered Systems

- Similar to storage mirroring
- Duplicate in-memory columns on another node
 - Enabled per table/partition
 - e.g. only recent data
 - Application transparent
- Downtime eliminated by using duplicate after failure

What Workloads Benefit From Database In-Memory



What is an analytic query?

Which products
give us our highest
margins?

Who are the top 10
sales reps in the north
west region this
month?

If I get a 20% discount
on widget A, how
much will our margins
improve?



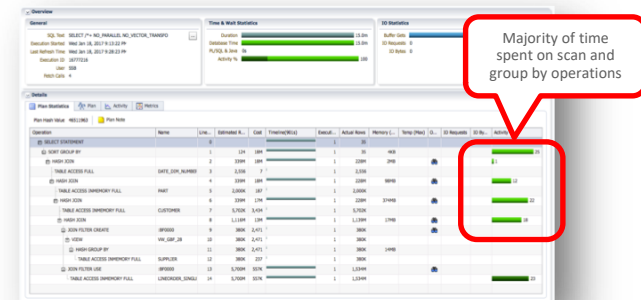
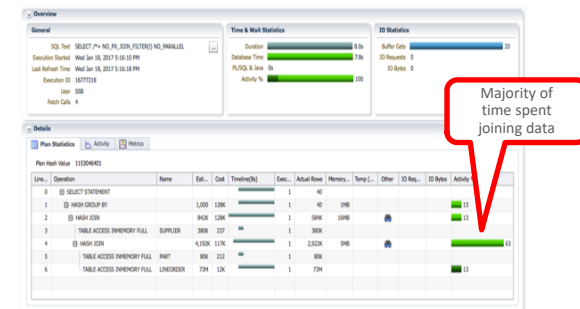
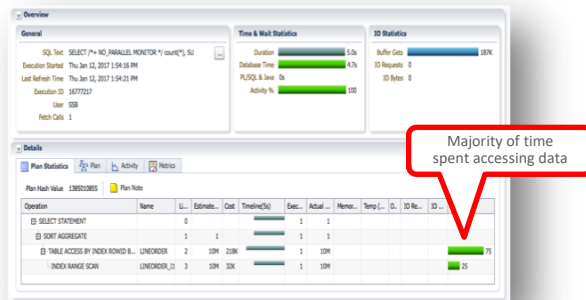
Queries that Benefit from Database In-Memory

Analytic Queries

- Return as few rows as possible
- Limit the number of columns accessed
- Use aggregation to return fewer values
- Use selective column predicates
- Use selective join conditions
- Limit the number of tables being joined
- Avoid complex SQL functions

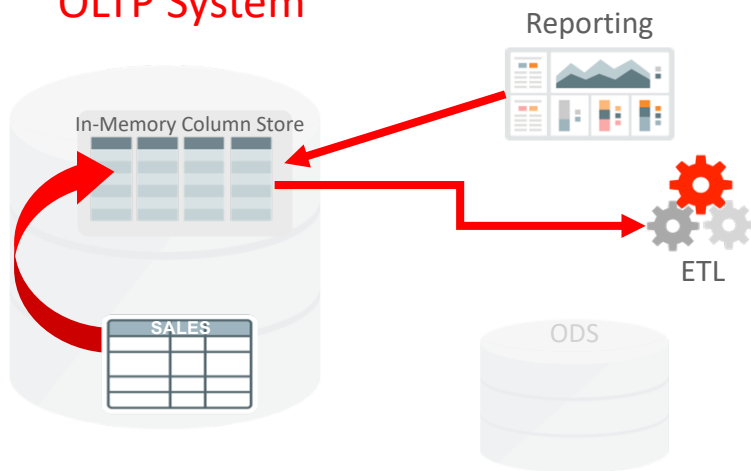
When Database In-Memory Helps

- For a non-trivial amount of rows and execution time:
 - Majority of time spent accessing data
 - Majority of time spent joining data
 - Majority of time spent on scan and group by operations



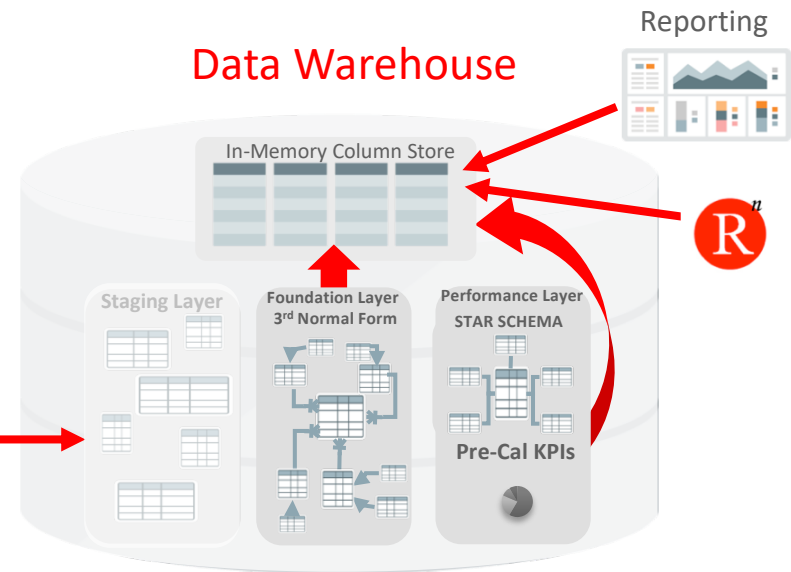
Where to use In-Memory

OLTP System



- Enables real-time reporting directly on OLTP data
- Speeds data extraction part of ETL process
- Removes need for separate ODS
- Speeds up mixed workload

Data Warehouse



- Star-schema and pre-calculated KPIs
 - Improves performance of dash-boards
- All or a subset of Foundation Layer
 - For time-sensitive analytics on 3rd normal form
- Staging/ETL/Temp not good candidates
 - Write once, read once

How to Compare Benefit

- Establish a baseline with no in-memory on the destination database
 - Ensure that the new environment runs the workload as well as the old environment
- Run with In-Memory enabled
- Ensure steady state – fully adapted plans, multiple executions
- Use time based comparison – SQL Monitor active reports
- Ensure no regressions – SQL Plan Baselines
- Remember Database In-Memory includes Hash Joins with Bloom filters and Vector Group By

See the Database In-Memory Implementation and Usage Whitepaper

How are customers using Database In-Memory



Database In-Memory References

AT&T WiFi – Data Warehouse



- Business Objects reports **100X** faster
- ETL processes improved by **50%** faster
- No changes to SAP Business Objects reports

Villeroy & Boch – SAP BW



- SAP BW COPA queries **30 – 33X** faster
- SAP Transaction list queries **4 – 4,800X** faster
- Avoided expensive & risky upgrade to S4/Hana

BOSCH – SAP CRM



- **Dropped** all custom indexes
- Analytic queries **2-20X** faster, DML **2-3X** faster
- No changes to application required

Die Mobiliar – Mixed Workload *Die Mobiliar*

- Analytic queries **50-200X** faster
- Database size **reduced** considerably
- Phase out of Netezza and mainframe systems

Database In-Memory Customers

Mankind Pharma – Mixed Workload



- Analytical reports **11x** faster
- Dropping indexes improved OLTP
- **90% reduction** in database size

Shanghai Customs – Mixed Workload

- Processes Clearance **43x** Faster
- Improves Declaration-Services Efficiency
- Reduced Costs

LION – SAP ERP



- Analytic queries **4X** faster
- Transactions **2X** faster
- Analytic queries now possible on 100s of Millions of Point-of-Sales Transactions

Lufthansa – Reporting Application



- Analytic queries up to **100x** faster
- Improved data ingest performance
- Reduction in database size

Where can I get more information



Additional Resources



Join the Conversation

- https://twitter.com/db_inmemory
- <https://twitter.com/TheInMemoryGuy>
- <https://blogs.oracle.com/In-Memory/>
- <https://www.facebook.com/OracleDatabase>
- <http://www.oracle.com/goto/dbim.html>

White Papers (otn.com)

- [Oracle Database In-Memory White Paper](#)
- [Oracle Database Implementation and Usage White Paper](#)
- [Oracle Database In-Memory Aggregation Paper](#)
- [When to use Oracle Database In-Memory](#)
- [Oracle Database In-Memory Advisor](#)

Videos

- [Oracle Database In-Memory YouTube Channel](#)
- [oracle.com](#)
[Powering the Real-Time Enterprise](#)
oracle.com/us/corporate/events/dbim/index.html
- YouTube - Juan Loaiza: [DBIM: What's new in 12.2](#)

Additional Questions

- In-Memory blog: blogs.oracle.com/In-Memory
- My email: andy.rivenes@oracle.com